EFFICIENTLY CALCULATING A BRANCH TARGET ADDRESS

CROSS REFERENCE TO RELATED APPLICATION

The present invention is related to the following U.S. Patent Application which is incorporated herein by reference:

Serial No. 10/082085 (now U.S. Patent No. 6,816,962) (Attorney Docket No. RPS20010178US1) entitled "Re-Encoding Illegal Op Codes Into a Single Illegal Op Code to Accommodate the Extra Bits Associated with Pre-Decoded Instructions" filed 2/25/02

TECHNICAL FIELD

The present invention relates to the field of instruction execution in computers, and more particularly to calculating a target address for a branch instruction upon fetching the branch instruction from an instruction cache without implementing adders while not substantially increasing the instruction length.

BACKGROUND INFORMATION

Program instructions for a microprocessor are typically stored in sequential, addressable locations within a memory. When these instructions are processed, the instructions may be fetched from consecutive memory locations and stored in a cache commonly referred to as an instruction cache. The instructions may then be retrieved from the instruction cache and executed. Each time an instruction is fetched from memory, a next instruction pointer within the microprocessor may be updated so that it contains the address of the next instruction in the sequence. The next instruction in the sequence may commonly be referred to as the next sequential instruction pointer. Sequential instruction fetching, updating of the next instruction pointer and execution of sequential instructions continues linearly through memory until an instruction, commonly referred to as a branch instruction, is encountered. A branch instruction